# Evolution Algorithm application in Neuron Network Music Genre Classification model of EEG

Liyao Zhang School of Computer Science, Australian National University <u>u6524995@anu.edu.au</u>

#### Abstract:

Recently, researchers interested in classifying music genre by discovering the pattern of the electroencephalogram (EEG). The method of neuron network (NN) is famous by discovering pattern by NN itself, however it is hard to balance the exploration and exploitation manually and find a best parameter manually. So, applying evolution algorithm becomes a suitable choose. In this research paper, we are purpose to build an evolution algorithm to generate the parameter automatically by using crossover and mutation to find suitable parameter setting to classifying music genre based on EEG.

#### **1.Introduction:**

Emotion and cultural beliefs contain in music which allow music to stimulus humans emotion. Recently, one research is to discover the pattern of electroencephalogram related to specific emotion [1]. The electroencephalogram is used to record the wave pattern of brain and we tries to classify these patterns into 3 different music genres. In this research we only focus on the brainwave in F7.



Channel location and Name [1]

### **1.1 Background:**

In previous research, we design a NN model which apply dropout process and CosineAnnealingLR improve the accuracy [3] however we did lots of manually adjustment in parameter. It is hard to try all the combination and balance the exploration and exploitation since in NN we have no idea about the inner connection between different parameter settings and the output accuracy. So, we do manually try instead to scientifically adjust. Evolution algorithm is an algorithm which uses crossover and mutation to discover the suitable setting more reasonably and automatically.

## **1.2 Dataset**

In the experiment, we use the subset of research paper "The effect of music on brain wave patterns" which collect the EEG data from 24 students, 13 male and 11 female. The original data set have 14 different channels, but we only have a channel F7 which is the left front lobe channel, so we are expected to collect different data from original paper.

# 2. Experiment method

#### 2.1 Pre-processing:

The whole dataset contains 576 data. I firstly clean the abnormal data. I define abnormal data as data who is 5 more times than std of the rest of data. I choose 5 more times because I do not want to clean too much data. 576 data is a small dataset for a NN model. Cleaning too much data will make the data become more inaccuracy. After the dataset was cleaned, I choose to apply normalization to decrease the difference between columns and rows. Finally, I randomly separate the dataset to test set which takes 10% of whole dataset and train set which takes 90% of train set and purpose to calculate validation accuracy with validation set which take 10% of train set.

## 2.2 Individual creation

I define individual as 1-layer or 2-layer NN model which contains 3 parameters which could be crossover and mutated, linear which is as known as the neuron node number between 3 and 30, active function and dropout rate. The active function will be randomly selected from Identity, RELU, Sigmoid and RELU6. The dropout rate will be between 0 and 0.2.

### 2.3 Initial population generalization

We first randomly generate 4 individual and randomly choose a pair of them and compare the validation accuracy. We will keep the one with greater accuracy as parent 1 and do the same process in 4 individuals to generate parent 2. Then do crossover in parent 1 and parent 2 to find parent 1\* and parent 2\*. After crossover, we do mutation in parent 1\* and parent 2\* to generate parent 1\*\* and parent 2\*\*. Then we add parent 1\*\* and parent 2\*\* into offspring. We will keep repeat this process until size population equals to define population [4].

### 2.4 Current population to offspring population transferring

Firstly, I will find the top 20% accuracies in current population and directly add them into offspring population. This step is purpose to keep the local best as known as exploitation. Then I will always randomly choose 2 from whole current population and let them do crossover and mutation to create 2 new individuals [2].

## 3. Results

I set the parameter, population size = 16, generation = 15, mutation probability = 1, x probability = 0.9 and receive the final population as:

Dept	Neuron	Dropout	Active	Mean
h	node		function	accurency
	number			
1	[28]	[0.19100116121832922]	['RELU6']	0.48
1	[22]	0	['RELU6']	0.48
2	[4, 7]	[0.0014713289729435838,	['Identity',	0.48
		0.06672773003941129]	'RELU6']	
2	[24, 21]	[0.0024480310087993777,	['Sigmoid',	0.48
		0.1497180324995631]	'Identity']	
2	[24, 21]	[0.0024480310087993777,	['Sigmoid',	0.48
		0.1497180324995631]	'Identity']	
2	[18, 6]	[0.00620689352797681,	['Sigmoid',	0.48
		0.03762237227787524]	'Identity']	
2	[18, 6]	[0.00620689352797681,	['Sigmoid',	0.48
		0.03762237227787524]	'Identity']	
1	[29]	[0.03296092255151428]	['Sigmoid']	0.48
2	[18, 6]	[0.00620689352797681,	['Sigmoid',	0.46
		0.03762237227787524]	'Identity']	
2	[24, 3]	[0.03528125875955823,	['Sigmoid',	0.46
		0.07141600652700912]	'Identity']	
1	[19]	[0.0015961652161779437]	['Identity']	0.46

2	[8, 3]	[0.07150779190016812,	['Sigmoid',	0.46
		0.14370763006709028]	'Identity']	
2	[8, 3]	[0.07150779190016812,	['Sigmoid',	0.46
		0.14370763006709028]	'Identity']	
2	[8, 3]	[0.07150779190016812,	['Sigmoid',	0.46
		0.14370763006709028]	'Identity']	
2	[30, 19]	[0.030552142599634704,	['RELU',	0.46
		0.13431016006649932]	'RELU6']	
2	[30, 19]	[0.030552142599634704,	['RELU',	0.46
		0.13431016006649932]	'RELU6']	

I randomly choose the one with the highest mean accuracy, Depth:1, Neuron node number: [28], Dropout: [0.19100116121832922], Active function: ['RELU6'], Mean accurency:0.48 and the accuracy of this in a test set is acc = 45.45%.

I also tried to set parameter with smaller mutation rate, population size = 20, generation = 10, mutation probability = 0.2, x probability = 0.9 and the final generation population is:

Dept	Neuron	Dropout	Active	Mean
h	node	_	function	accurency
	number			
2	[26, 18]	[0.19041104755748547,	['Sigmoid',	0.48
		0.02816593587201601],	'RELU6']	
2	[26, 18]	[0.18687357170501823,	['Sigmoid',	0.48
		0.02816593587201601]	'RELU6']	
2	[26, 18]	[0.19035423565584234,	['Sigmoid',	0.48
		0.047557029479989806]	'RELU6']	
2	[26, 18]	[0.18687357170501823,	['Sigmoid',	0.48
		0.02816593587201601]	'RELU6']	
2	[26, 18]	[0.19035423565584234,	['Sigmoid',	0.48
		0.047557029479989806]	'RELU6']	
2	[25, 18]	[0.19010326126105434,	['Sigmoid',	0.48
		0.04542075397403759]	'RELU6']	
2	[26, 18]	[0.18687357170501823,	['Sigmoid',	0.48
		0.02816593587201601]	'RELU6']	
2	[26, 18]	[0.18687357170501823,	['Sigmoid',	0.48
		0.02816593587201601]	'RELU6']	
2	[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
		0.04406528753538387]	'RELU6']	
2	[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
		0.04406528753538387]	'RELU6']	
2	[30, 16]	[0.2, 0.2]	['Identity',	0.46
			'Sigmoid']	
2	[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46

	0.04406528753538387]	'RELU6']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[30, 16]	[0.2, 0.2]	['Identity',	0.46
		'Sigmoid']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[25, 17]	[0.18985127422743014,	['Sigmoid',	0.46
	0.04406528753538387]	'RELU6']	
[30, 16]	[0.2, 0.2]	['Sigmoid',	0.4
		'Sigmoid']	
	[25, 17] [25, 17] [25, 17] [30, 16] [25, 17] [25, 17] [25, 17] [30, 16]	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	0.04406528753538387] 'RELU6']   [25, 17] [0.18985127422743014, 0.04406528753538387] ['Sigmoid', 'RELU6']   [25, 17] [0.18985127422743014, 0.04406528753538387] ['Sigmoid', 'RELU6']   [25, 17] [0.18985127422743014, 0.04406528753538387] ['Sigmoid', 'RELU6']   [30, 16] [0.2, 0.2] ['Identity', 'Sigmoid']   [25, 17] [0.18985127422743014, 0.04406528753538387] ['Sigmoid']   [25, 17] [0.18985127422743014, 0.04406528753538387] ['Sigmoid', 'RELU6']   [30, 16] [0.2, 0.2] ['Sigmoid', 'Sigmoid']

I randomly choose the one with the highest mean accuracy, Depth:1, Neuron node number: [26, 18], Dropout: [0.19041104755748547, 0.02816593587201601], Active function: ['Sigmoid', 'RELU6'], Mean accurency:0.48 and the accuracy of this in a test set is acc = 40.0%.

# 4.Discussion:

## 4.1 Validation accuracy and test accuracy

The validation accuracy is slightly higher than test accuracy which is acceptable. Comparing with the accuracy of author, 97% accuracy is much higher than 40% [1]. Beside our dataset is small and only contain F7's feature, the author may build a more complex NN model. Our EA only do 1-layer or 2-layer's NN model generation. It could do more layer and more generation, but the calculation time will be extreme long, so this is the reason why I choose 1-layer or 2-layer's NN model to support this method is valid.

#### 4.2 Comparing different mutation rate

After that I will compare two different mutation rates. The one which choose 1 as mutation rate which means 100% mutation rate shown the better result. I think it is because of my design of population generalization. The divided each generalization of generation into two separate parts, exploration part and exploitation part. The exploration part which is affected by mutation rate is purposed to explore to find the other possible local maximum and global

maximum. It also avoids all the population gather around one peak. The exploitation part is guaranteeing the maximum and peak is always be saved. So, this cause the exploration part could be unlimited to keep a quick speed to find as much as possible combination to find better maximum in a small generation number. The exploitation part makes sure local maximum always save that before the global maximum find. Since exploration part and exploitation part is separated, the mutation changes do not affect exploitation part which means a greater mutation rate will only increase the speed of trying and will not affect current maximum at all.

#### 5 Conclusion and future work

The model accuracy of data does not perform as well as the J. S. Rahman's article mentions because we only use F7 data, however we does develop an automatic method to find best parameter for building a NN model. In future task, we could add more locations data and increase the generation size and layer number to have a better model. We also could apply Q training to guide the EA to make mutation is guided. This will increase the speed and efficiency of mutation to make the better parameter could be appear in a small generation.

### Reference

- [1]J. S. Rahman, T. G. (2020, July). Brain Melody Informatics: Analysing Effects of Music on Brainwave Patterns. 2020 International Joint Conference on Neural Networks (IJCNN), pp. pp. 1-8.
- [2]Liu, Y., Sun, Y., Xue, B., & Zhang, M. (2020, July 24). Evolving Deep Convolutional Neural Networks for Hyperspectral Image Denoising. 2020 International Joint Conference on Neural Networks (IJCNN).
- [3]Mishra, S. Y. (2019, May). Improving image classifiers for small datasets by learning rate adaptations. 2019 16th International Conference on Machine Vision Applications (MVA), pp. pp. 1-6.
- [4]Sun, Y., Xue, B., Zhang, M., & Yen, G. G. (2019, May 10). Evolving Deep Convolutional Neural Networks for Image Classification. *IEEE Transactions on Evolutionary Computation*, pp. 394 - 407.